



#16 WJ
4-30-04
AF/36-28
PATENT
Attorney Docket No. 202231

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:

Timothy Charles Sowell

Art Unit: 3628

Application No. 09/418,943

Examiner: Nguyen, Nga B.

Filed: October 15, 1999

For: Distributing and Billing Software
According to Customer Use of Program
Modules

TRANSMITTAL OF
APPELLANTS' APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED
APR 13 2004
GROUP 3600

Dear Sir:

In accordance with 37 CFR 1.192, appellants hereby submit Appellants' Brief on Appeal in triplicate.

The items checked below are appropriate:

1. Status of Appellants

This application is on behalf of ☒ other than a small entity or ☐ a small entity.

2. Fee for Filing Brief on Appeal

05/03/2004 LWALDEN 00000008 121216 09418943

01 FC:1252

420.00 DA

Pursuant to 37 CFR 1.17(c), the fee for filing the Brief on Appeal is for: ☒ other than a small entity or ☐ a small entity.

Brief Fee Due \$330.00

3. Oral Hearing

☒ Appellants request an oral hearing in accordance with 37 CFR 1.194.

CERTIFICATE OF MAILING

I hereby certify that this document (along with any documents referred to as being attached or enclosed) is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date: April 5, 2004

Julia L. Rivera

In re Appln. of Sowell
Application No. 09/418,943

4. Extension of Time

- ☒ Appellants petition for a two-month extension of time under 37 CFR 1.136, the fee for which is \$420.00.
- ☐ Appellants believe that no extension of time is required. However, this conditional petition is being made to provide for the possibility that appellants have inadvertently overlooked the need for a petition and fee for extension of time.

Extension fee due with this request: \$420.00

5. Total Fee Due

The total fee due is:

Brief on Appeal Fee	\$330.00
Request for Oral Hearing	\$290.00
Extension Fee (if any)	\$420.00

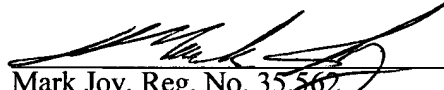
Total Fee Due: \$1,040.00

6. Fee Payment

- ☐ Attached is a check in the sum of \$
- ☒ Charge Account No. 12-1216 the sum of \$1,040.00. A duplicate of this transmittal is attached.

7. Fee Deficiency

- ☒ If any additional fee is required in connection with this communication, charge Account No. 12-1216. A duplicate copy of this transmittal is attached.


Mark Joy, Reg. No. 35562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson Avenue
Chicago, Illinois 60601-6780
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: April 5, 2004



PATENT
Attorney Docket No. 202231

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Timothy Charles Sowell

Art Unit: 3628

Application No. 09/418,943

Examiner: Nguyen, Nga B.

Filed: October 15, 1999

For: Distributing and Billing Software
According to Customer Use of Program
Modules

RECEIVED
APR 13 2004
GROUP 3600

APPELLANT'S APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In support of the appeal from the final rejection dated September 8, 2003, Appellant now submits this Appeal Brief.

(1) *Real Party In Interest*

The patent application that is the subject of this appeal is owned by Invensys Systems, Inc.

(2) *Related Appeals and Interferences*

There are no appeals or interferences that are related to this appeal.

(3) *Status of Claims*

Claims 1-12, 14-44, 53-55 and 74-77 are pending in this application, and *all are appealed*.

Claims 45-52 and 56-73 were previously withdrawn from consideration in response to a restriction requirement. Claim 13 was previously deleted.

Pending claims 41, 54 and 55 were last amended in Appellant's Amendment dated November 5, 2001.

04/12/2004 CNGUYEN 00000024 121216 09418943
01 FC:1402 330.00 DA

Pending claims 1, 16, 31, 33, 34, 42, and 53 were last amended and claims 74-77 were added in Appellant's Amendment dated August 5, 2002.

(4) *Status of Amendments*

No further amendments have been submitted in association with this appeal.

(5) *Summary of Invention*

The appealed claims of the present invention generally relate to methods and a software module structure that facilitate charging customers for use of software modules distributed to the customers' sites. Claims 53-55 are directed to a particular arrangement of program/data within a software module that facilitate executing the claimed use-based methods for charging customers for using distributed software modules containing an object from which instantiated objects are rendered.

In accordance with claim 1, corresponding generally to the steps summarized in FIG. 7, a method for charging customers for using software object instances is recited. A use-based pricing scheme is established (e.g., step 100 described at page 19). Thereafter, a set of software modules, including at least one object class, are distributed to customers (e.g., steps 102 and 104 described at pages 19 and 20). The software on the users' systems monitor usage of the software by the users (e.g., step 112 described at pages 21-23). The monitoring step, as well as the preceding steps, contain detailed examples of creating object instances, monitoring use of the object instances created from an object class (e.g., step 110 described at page 21), and thereafter charging customers (e.g., step 112 where credits are deducted from previously purchased credits stored in customer account during step 108) based upon the monitored usage of the object instances.

With regard to claim 2, multiple instances are created from the software module (containing an object class), and use of the software is measured according to detected instances during the monitoring step (see, e.g., step 112, "daily" and "lifetime" modes described at page 22).

With regard to claim 3, the use is periodically monitored (e.g., repetition of steps 112 and 114 described at pages 22 and 23).

With regard to claims 4 and 36, usage is measured by each day that an object instance is active and a user is charged a daily rate (e.g., "daily" usage described at page 22, lines 10-18).

With regard to claim 5, a demonstration mode is provided wherein object instances are executable free of charge (e.g., step 106 described at page 20, lines 20-25).

With regard to claim 6, the claimed method includes maintaining a single agreement governing use of the object instances created from the set of software modules for an enterprise (e.g., step 110, described at page 21).

With regard to claim 7, termination dates of time-limited object instances are monitored (e.g., step 106, at page 20, line 28 to page 21, line 2).

With regard to claims 8, 26, and 41, upcoming expiration dates of object instances are monitored and warnings are issued (see, e.g., page 9, lines 20-23; page 15, lines 18-20).

With regard to claim 9, the claimed method includes maintaining an account of credit units and during the charging step, the account credits are decremented based upon the monitoring step (e.g., step 112 described at page 21, lines 23-27).

With regard to claim 10, a report is generated summarizing use of software modules at the customer site (e.g., step 114 described at page 23, lines 8-10).

With regard to claim 11, charging is based upon registered uses of a software module (see, step 112 described at page 22, line 29 to page 23, line 4).

With regard to claims 12 and 32 use is measured according to *execution* of created instances (e.g., daily usage mode described in step 112 described at page 22, lines 10-28).

With regard to claim 14, the software module is an object class for creating an application engine object (e.g., page 24, lines 22-27).

With regard to claim 15, the software module is an object class for creating a view engine object (e.g., page 22, lines 12-15).

With regard to claim 16, the monitoring step (112) determines a time duration that an object instantiated from a software module is active (see, demo mode described at page 22, lines 24-26).

With regard to claims 17, 34, and 77 the monitoring step comprises registering execution of an instance that tracks throughput of a process (see, page 18, lines 9-14).

With regard to claim 18, individual ones of the set of software modules are individually priced (see, Fig. 6, fields 92 and 94, described at page 18, lines 1-18).

With regard to claim 19, the set of software modules includes at least a first software module supplied by a third party vendor and the method further comprises compensating a third party vendor based upon a use by a customer of the first software module determined during the monitoring step (see, step 114 described at page 23, lines 11-16).

With regard to claims 20 and 21, the software modules are transmitted via a network, and more particular an Internet, connection (see, distribution function 22, at page 8, lines 24-27 and steps 102 and 104 of FIG. 7).

With regard to claim 22, usage information is reported to a software brokerage facility. (see, e.g., licensing function 24, at page 8 line 28 to page 9, line 2 and brokerage 10 described at page 10, lines 21-23).

With regard to claim 23, reporting includes identifying a location of an instance (see, page 15, lines 8-11).

With regard to claim 24, a failure by a license manager to report to a software brokerage facility is detected, and in response a communication failure is recorded at a central licensing facility (see, page 14, lines 26-30).

With regard to claim 25, monitoring includes storing use information in summary format in a database (see, step 112 and usage register 76 described at page 21, lines 23-30).

With regard to claims 27, 28, and 40 the software modules relate to industrial manufacturing automation/information software (see, e.g., process control (automation) objects and process data viewing (information) objects described in FIG. 8, page 11, line 28 to page 12, line 4; page 18, lines 6-15).

With regard to claim 29, an agreement is maintained for governing use of instances created from the set of software modules for an enterprise wherein the instances comprise both lifetime billed and use-based billed instances (see, page 19, lines 7-10; and page 22, lines 10-28).

With regard to claim 30, configuration tools are provided that enable a user to create customized instances from the software modules (see, page 20, lines 4-8).

Claim 31 recites a method for vending software in the form of software modules via electronic commerce channels. The recited method includes maintaining an electronic commerce site including a software module selection interface that enables a customer to request a software module for use at a customer site, wherein the software module comprises at least one object class from which objects are instantiated on a customer system. (See, step 102 described at the bottom of page 19). A software module management framework is provided to the customer for installation at a customer site, wherein the management framework includes components for registering use of the software module at the customer site (see, the license manager 66 described at page 14, lines 15-28). Thereafter, the customer is charged based upon registered use of software modules, wherein software module usage is

measured according to object instances created from the at least one object class (e.g., step 112, described on page 21, line 23 to page 23, line 4, where credits are deducted from previously purchased credits stored in customer account during step 108).

With regard to claims 33, 74, 75 and 76 usage of the software is based upon detecting *creation* of instances of an object at the user's site during the monitoring step (e.g., step 112, see "lifetime mode" described at page 22).

With regard to claim 35, the module management framework (e.g., license manager 66) supports creation of instances from software modules at the customer cite having differing use modes including at least: a lifetime mode and a use-based mode, and wherein said method comprises the further step of registering execution of instances operating in the use-based mode (see, e.g., step 112 described at pages 21-23).

With regard to claim 36, usage is measured in days, and an instance operating in use-based mode is registered each day in which the instance is executed (see, step 112 at page 22, lines 10-18).

Claim 37 recites a method for charging customers for use of software. The method includes the step of providing a set of individually identifiable units of software including at least one object class from which objects are instantiated on a customer system (e.g., steps 102 and 104 described at pages 19 and 20). The downloaded units are individually priced (see, description of modules provided in FIG. 5, including daily 92 and lifetime 94 rate fields). Authorizing use of the executable software is performed (e.g., installing a licensing agreement described at step 110 described at page 21, lines 14-22). Thereafter, a customer is charged (e.g., step 112 where credits are deducted from previously purchased credits stored in customer account during step 108) based upon use of selected ones of the set of individually identifiable units of software, and software usage is measured according to object instances created from the at least one object class.

With regard to claim 38, the authorizing step comprises transmitting a license file containing code enabling use by the customer of the executable software (see, page 16, line 27 to page 17, line 3).

With regard to claim 39, the method further includes integrating self-monitoring process software (see, page 18, lines 1-20) within the executable software, and registering use of the executable software by the self-monitoring process (e.g., description of step 112 at page 22, lines 3-28).

Claim 42 recites a method for charging customers for use of software. The method includes first providing a set of software modules for software customers, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system (e.g., steps 102 and 104 described at pages 19 and 20). Second a software licensing facility is provided that includes a brokering facility (software brokerage 10) through which software customers pay for software execution units, and wherein the brokering facility includes a set of software customer accounts (see, software brokerage 10 described at page 6, lines 7-30, page 10, lines 5-18). Customer accounts are charged a number of software execution value units based upon the value of software modules utilized by a customer (page 10, lines 5-18), and wherein software module usage is measured according to object instances created from the at least one object class (see, FIG. 7, described at pages 17-23 describing an "object-based" use charging scheme).

With regard to claim 43, customer charging is performed by an automated billing process (see, step 112 described at pages 21-23).

With regard to claim 44, an on-line customer interface provides an interface enabling users to download software modules from a remote location (see, e.g., page 8, lines 24-27, and page 19, line 19 to page 20, line 11).

With regard to claim 53, a memory containing a software module structure facilitating automated distribution of software to customers is described in FIG. 6 (see description of billed software modules 72 described at , page 17, line 16 to page 19, line 4). The recited software module structure includes a supplier identification (within vender details 86), a product description (within vender details 86 and module class field 88), a billing definition (usage rate 92, lifetime value 94); and an executable program module including at least one object class from which instantiated objects are rendered (data segment 98).

With regard to claim 54, the billing definition within the software module structure includes a usage rate (92), and with regard to claim 55, a lifetime rate (94).

(6) Issues

The following general issues have been raised by the final rejection of the above summarized pending claims.

1. Whether claims 53-55 are unpatentable under 35 U.S.C. §101 as being directed to non-statutory subject matter.

2. Whether claims 1-12, 14-44, 53-55 and 74-77 are unpatentable under 35 U.S.C. §103(a) as obvious over Archibald et al. U.S. Pat. No. 5,825,883.

(7) *Grouping of Claims*

The claims do not stand or fall together. Rather, each one stands or falls on its own for at least the reasons set forth herein below. However, for purposes of simplifying this appeal, the claims are grouped as follows for the argument set forth herein below:

Group I: 1, 9-11, 18-22, 25, 29, 31, 35, 37, 39, 42, 43, and 44
Group II: 2
Group III: 3
Group IV: 4, 36
Group V: 5
Group VI: 6
Group VII: 7
Group VIII: 8, 26, and 41
Group IX: 12, 32, 33, 74-76
Group X: 14
Group XI: 15
Group XII: 16
Group XIII: 17, 34, 77
Group XIV: 23
Group XV: 24
Group XVI: 27, 28, and 40
Group XVII: 30
Group XVIII: 38
Group XIX: 53-55

The large number of groups was necessitated by the absence of a number of recited claim elements that simply are not disclosed or even remotely suggested by the Archibald reference. The remaining claims cannot be grouped due to their separate grounds for patentability set forth herein below. Furthermore, notwithstanding Appellant's grouping of the claims, *Appellant incorporates by reference, and explicitly reserves the right to reassert, each and every ground set forth in any preceding Office Action response* to the extent needed to distinguish the invention from the prior art.

(8) *Argument*

In summary of Appellant's argument on appeal, Archibald et al. U.S. Patent 5,825,883 neither discloses nor suggests the claimed object usage-based software monitoring/billing scheme embodied in each of the currently pending claims involved in this appeal. The Final Office action goes beyond the disclosure of the Archibald '883 patent and seeks to modify its disclosed teachings in a way that is neither suggested nor disclosed in the prior art. The Final Office Action concedes that Archibald makes no mention of object-oriented program constructs. However, the Final Office Action then proposes that one skilled in the art would have been motivated to modify the teachings of Archibald, based solely upon the existence of object-oriented programming at the time of the invention, to render Appellant's invention recited in the independent claims. To the contrary, there is no teaching or suggestion in the cited prior art for such general modifications to Archibald, let alone the particular modifications needed to render Appellant's claimed invention.

Furthermore, with regard to the rejection of claims 53-55 as claiming non-statutory subject matter, Appellant respectfully asserts that these claims are directed to patentable "functional descriptive material" described within the MPEP at 2106 IV.B.1 and by the Federal Circuit in *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ.2d 1031, 1035 (Fed. Cir. 1994). In contrast to such non-patentable subject matter such as music, books, and videos, the claimed memory contains a software module structure including a combination of functional fields (including at least one object class from which executable object instances are created) that are arranged to provide particular information supporting object usage-based billing of an executable program including at least one object class from which instantiated objects are rendered. The claimed invention fits squarely within the defined patentable subject matter defined in both the MPEP and the Federal Circuit's *In re Lowry* decision.

A. The Section 101 Rejection of claims 53-55

Appellant traverses the rejection, in **Sections 5 and 6** of the Final Office Action, of claims **53-55** as being directed to non-statutory subject matter. The Final Office Action states that the claims recite "non-functional descriptive material." However, as Appellant pointed out in previous responses, the claimed invention is directed to "functional descriptive material." The MPEP, at 2106 IV.B.1, entitled: Non-statutory Subject Matter, states as follows:

...Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of *data structures and computer programs* which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data. (emphasis added)

Section 2106 IV.B.1 later adds that "when functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ.2d 1031, 1035 (Fed. Cir. 1994)."

Using the MPEP and *Lowry* as guidance, claims 53-55 represent patentable subject-matter. With regard to the first part of the two-part test summarized above, the recited claim elements, including "an executable program module including at least one object class from which instantiated objects are rendered," are directed to *functional* descriptive material. Furthermore, the claimed invention (including a combination of object classes and charge rates for use of the objects), taken as a whole, performs the function of facilitating automated distribution and charging (for use of distributed objects) of software to customers. The claimed invention, explicitly reciting "an executable program module" is clearly functional. Appellant requested an explanation of why the preamble and recitation of executable software in claims 53-55 does not meet the "functional" requirement for descriptive material. The Final Office Action does not address this request, nor does it analyze the claimed subject matter in view of *Lowry*.

With regard to the second part of the patentability test, the preamble recites "a memory containing a software module." Such memory, read in light of the last element (i.e., "an executable program module including at least one object class"), meets the second

part of the test summarized above – that the claimed functional descriptive material be recorded on some computer-readable medium. In this regard, the Final Office Action appears to measure the claims against the *Beauregard* claim standard (steps of a method incorporated into computer-executable instructions stored on a computer-readable media). As mentioned previously above, claims 53-55 are drafted according to the guidelines of *Lowry* which described patentable data structures. If the Section 101 rejection of claims 53-55 is not withdrawn, then Appellant respectfully requests a specific explanation of how this two-part test set forth in the MPEP, and based on *Lowry*, is not met by currently pending claims 53-55.

B. The Section 103 Rejection of claims 1-12, 14-44, 53-55 and 74-77

Appellant furthermore traverses the rejection, in **Sections 7 and 8** of the Final Office Action, of claims 1-12, 14-44, 53-55 and 74-77 under 35 U.S.C. Section 103(a) as being unpatentable over Archibald U.S. Patent No. 5,825,883 in view of specified and unspecified prior art. Appellant addresses each of the rejections in the order they arise in the Final Office Action.

Group I: Claims 1, 9-11, 18-22, 25, 29, 31, 35, 37, 39, 42, 43, and 44

The mere fact that object-oriented programming existed at the time Appellant conceived the current invention does not support the obviousness conclusion reached within the Final Office Action. Appellant's claimed invention goes beyond merely claiming that customers are charged for using applications that contain objects. Instead, the claims recite that object instances are created from object classes that are distributed to customer systems. The customers are thereafter charged in accordance with the monitored use of the created object instances. The Archibald '883 patent in no way suggests object usage-based charging, wherein a customer is charged based upon monitored use of objects created from one or more object classes contained within a downloaded software module on a customer's system as recited in each of the independent method claims. Nor does Archibald disclose the software module structure, recited in claim 53, that facilitates such customer charge arrangement. In fact, Archibald does not even mention a single object-oriented programming construct (object, instance, class, etc.) throughout its entire specification, or how its metering system would carry out charging for use of object instances in the event that its applications did, indeed, comprise object classes. For at least this reason, the claims cannot be obvious over the prior art presently known to Appellant. The obviousness rejections will be further addresses, with reference to particular claims, herein below.

In addition to the general argument set forth above, Appellant submits that the rejection of claim 1, in the Final Office Action is based upon construing a number of object-oriented program environment terms, recited in claim 1, in a manner that is inconsistent with well-established meanings for those terms in the art. In particular, the Final Office Action construes object-oriented program environment terms, such as "object class," "object instances," "instantiated," etc., in a manner that is inconsistent with Appellant's specification and the knowledge of those skilled in the computer software art.

Nowhere does Archibald disclose or suggest measuring usage according to object instances instantiated from a distributed object class as recited in claim 1. As an initial matter, the Final Office Action, at page 5, explicitly concedes that Archibald does not mention the terms "class" or "object." Thereafter, the Office Action recites/reference a passage, extracted from a reference entitled: "Data abstraction and structures using C++", defining object classes, object instances, etc. While incomplete, the citation of the reference reinforces that claim 1, as well as the other independent claims, is clearly directed to an object-oriented program environment. However, rather than concede that Archibald does not suggest the "object-creation-based" accounting of customer usage recited in claim 1, the Final Office Action, proceeds to apply the terms "object class" and "object instances" to the disclosure of Archibald in a way (i.e., non-object-oriented) that is inconsistent with Appellant's specification and claims, and the cited definition from the "C++" reference.

Appellant's claim 1 recites object-oriented program environment constructs as well as a particular way in which to utilize such constructs to carry out a use-based software distribution/billing method. As suggested by the "C++" reference cited at page 5 of the Final Office Action, an object class is a template in an object-oriented (e.g., C++) program environment from which object instances are created (instantiated). Appellant's claim 1 specifically recites distributing an object class (template) from which object instances are created (instantiated). As recited in claim 1, charging a customer is based upon usage of the distributed object class as measured by *object instances created from the distributed object class*. Therefore, even if the applications of Archibald were to contain objects, the particular way in which the objects are created from distributed classes, and their use monitored, is neither disclosed nor suggested in Archibald.

Furthermore, the Final Office Action, at the bottom of page 5 seeks to identify object classes and instances, as recited in claim 1, in the Archibald reference. The Final Office Action assigns the object-oriented program environment labels "class" and "instance" to elements in

Archibald, even though nothing in Archibald's disclosure suggests that its distribution scheme is based upon such object-oriented program environment constructs. Appellant repeats that Archibald neither discloses nor suggests that its distribution scheme is indeed performed in an object-oriented environment. Thus, the rejection of claim 1 is improper for at least the reason that Archibald is not directed to an object-oriented program environment, and therefore cannot suggest the recited object instance-based usage measurement scheme, and should be withdrawn.

For purposes of the appeal, Appellant, for the reasons set forth hereinabove, likewise traverses the rejection of independent claims 31, 37, 42, and 53 (discussed separately herein below with reference to Group XIX) that are also directed to object-oriented program environment constructs that are neither disclosed nor suggested in Archibald. Appellant addresses the Final Office Action's individual grounds for the rejection of the dependent claims herein below.

Group II: Claim 2

Appellant has appealed the rejection of **claim 2** that recites "a customer creates a number of instances from a software module, and use is measured according to instances detected" Archibald does not disclose monitoring the creation of instances (e.g., object instances created from an object class) of an item. Archibald does not even disclose creation of such instances. Instead, the cited portion of Archibald, at col. 6, lines 48-60, merely discloses a meter module generating use information (*i.e., length of use*). The reference to "i.e., length of use" at line 55, not as an example, but rather as "in other words," does not suggest a use-based charging scheme based upon detecting created instances. It is further iterated that Archibald does not suggest usage based upon object-oriented "instances" created from the claimed object class.

Group III: Claim 3

Appellant has appealed the rejection of **claim 3**. Archibald discloses that usage information is determined. It does not disclose *how* that information is obtained. Nowhere does Archibald disclose instances created from a software module that are "periodically *accessed* to determine use." The Archibald does not disclose or suggest the recited element, and Appellant respectfully requests identification of where Archibald, at col. 12, lines 33-42, teaches periodically *accessing the instances* to determine their use.

Group IV: Claims 4 and 36

With regard to the rejection of **claim 4 and claim 36**, Appellant does not challenge that charging a daily rate for use of software is not known. Rather, Appellant traverses the rejection of claims 4 and 36 for at least the reason that the prior art does not disclose or suggest the object instance-based daily monitoring/charging steps recited in claim 4. In summary, the prior art neither discloses nor suggests charging for use of created object instances on a daily basis.

Group V: Claim 5

With regard to the rejection of **claim 5**, Appellant agrees that Archibald contemplates a "trial use." However, Archibald does not disclose that the "trial use" option is associated with the recited "demonstration mode" of an object instance. In other words, the object instance has a particular mode of operation that is associated with a demonstration state for the object instance.

Group VI: Claim 6

Claim 6 recites "a single agreement governing use of instances" created from software modules. However, Archibald discloses a license that appears to apply to each downloaded copy of digital content rather than instances that are created from the downloaded digital content. Furthermore, the Office Action appears to be misconstruing the phrase "governing use of instances created from the set of software modules" in seeking to apply the disclosure of col. 7, lines 40-67 of Archibald to the claimed invention.

Group VII: Claim 7

The rejection of **claim 7** is appealed for at least the reason that Archibald does not disclose "*instances derived from a software-module.*" Furthermore, Archibald teaches termination when an amount (rent to own price) is reached rather than a termination date.

Group VIII: Claims 8, 26 and 41

Appellant objects to the rejection of **claims 8, 26 and 41** including the assertions that the recited elements are both well known in the art and that incorporation of such elements into Archibald is suggested by the prior art. Appellant is unaware of any prior art reference disclosing the recited issuance of a warning/reordering reminder in response to detecting an upcoming expiration date, issuing a re-order reminder or informing a user of a need to reorder credits in the context of the recited invention. In the event that this rejection is not withdrawn,

In re Appln. Of Timothy Charles Sowell
Application Number: 09/418,943

Appellant requests provision of a reference showing such a teaching in the prior art. At such time Appellant will assess whether such reference, taken in combination with the teachings of Archibald, renders claims 8, 26 and 41 obvious.

Group IX: Claims 12, 32, 33, 74, 75 and 76

Appellant traverses the rejection of **claims 12, 32, 33, 74, 75 and 76** for at least the reasons provided herein above regarding the independent claims, and furthermore because these claims recite specific bases (when an object is created/executed) for monitoring/charging for use of object instances created from a class object that are neither suggested nor disclosed in Archibald.

Group X: Claim 14

Appellant respectfully submits that, with regard to the rejection of **claim 14** Archibald does not even disclose downloading an object class from which the specified objects are created/instantiated. Appellant specifically traverses the Office Action's assertion that "object classes" in general, and in particular "application engine objects" are disclosed or suggested by Archibald.

Group XI: Claim 15

Appellant respectfully submits that, with regard to the rejection of **claim 15** Archibald does not even disclose downloading an object class from which the specified objects are created/instantiated. Appellant specifically traverses the Office Action's assertion that "object classes" in general, and in particular "view engine objects" are disclosed or suggested by Archibald.

Group XII: Claim 16

Appellant appeals the rejection of **claim 16** in view of the previous amendment to the claim indicating that the monitored duration is that of an active object instantiated from the object class within the software module. Archibald does not disclose such object instances.

Group XIII: Claims 17, 34, and 77

Appellant appeals the rejection of **claims 17, 34 and 77**. Each of these claims is directed to a particular monitoring implementation that comprises "registering execution of an instance

that tracks throughput of a process" and this indirectly measures value created by a process that uses the software modules. Appellant respectfully submits that nothing in column 5, line 65 to col. 6, line 12 of Archibald (cited as the basis for rejecting the claims) even remotely discloses tracking process throughput or the particular method step recited in claim 17. The Final Office Action's further explanation does not address the "throughput of a process" element of claim 17.

Group XIV: Claim 23

Appellant traverses the rejection of **claim 23**. The claim element recites "identifying the location of an instance created from a software module" obtained by a customer. Appellant respectfully submits that Archibald, col. 10, lines 1-13 does not disclose identifying the location of an object instance created from a software module obtained by a customer. The subsequent explanation by the Final Office Action does not provide any further insight as to how Archibald discloses the recited elements of claim 23 pertaining to identifying a location of a software module.

Group XV: Claim 24

Appellant traverses the rejection of **claim 24**. Claim 24 is directed to a failure by a license manager (maintained at a customer's site) to communicate usage of software to a software brokerage. The failure is reported to a central licensing facility. In contrast, the Archibald reference discloses a failure by an authority (e.g., a software brokerage) to properly communicate to a customer site. Archibald's failure is a communication failure in the *opposite* direction of the element recited in claim 24.

Group XVI: Claims 27, 28 and 40

Appellant traverses the rejection of **claims 27, 28 and 40**. Archibald, at col. 2, lines 37-44 cited in the Final Office Action, does not disclose or even remotely imply that the downloaded software relates to industrial manufacturing (automation/information) software. If anything, Archibald suggests that the downloaded digital content comprises individual consumer items such as articles, music, books, individual consumer software, etc. – not the type of software utilized to run industrial processes and record data (information) from such processes.

Group XVII: Claim 30

Appellant traverses the rejection of **claim 30**. In particular, though configuration tools were indeed well-known at the time of the invention, Appellant respectfully submits that the prior art does not suggest providing such tools to customize instances created from software modules in a method that includes charging a customer based upon monitored use of software modules. The cited reference and explanation of configuring a software program does not meet the claim limitations directed to customizing object instances. In the event that this rejection is not withdrawn, Appellant requests provision of a reference showing such a teaching in the prior art. At such time Appellant will assess whether such reference, taken in combination with the teachings of Archibald, renders claim 30 obvious.

Group XVIII: Claim 38

Appellant traverses the rejection of **claim 38**. Archibald, at col. 6, lines 33-47, cited in the Final Office Action, merely discloses use identification information (in the meter data file), but does not suggest that this file is used to *enable* operation of executable software nor that it is transmitted during an authorization step. Appellant further traverses the assertion that col. 11 – col. 12 of Archibald discloses the recited transmitting a license file enabling use of the software.

Group XIX: Claims 53-55

Turning to the rejection of **claims 53-55**, Appellant submits that the presently claimed invention comprises a defined article of manufacture wherein particular information is logically bundled within a single module that facilitates efficient and accurate marketing, distribution, and accounting of use by customers of software modules. Appellant traverses the rejection of claims 53-55 as obvious for at least the reason that Archibald does not disclose a single module containing the recited object class.

Finally, Appellant reserves the right to traverse any of the dependent claims for at least the reason that the rejections require an obviousness analysis since the cited Archibald reference, in each rejected claim, does not include at least one recited element.

(9) *The Appealed Claims*

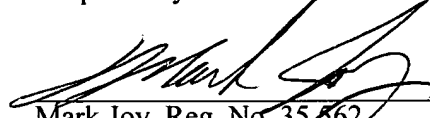
The appealed claims are set forth in the Appendix attached hereto.

In re Appln. Of Timothy Charles Sowell
Application Number: 09/418,943

Conclusion

In the Final Office Action preceding this appeal, there has been an absence of recognition that the claims require measuring use of software based upon object instances created from one or more object classes provided in distributed software modules. With regard to the rejection of claims 53-55 as reciting unpatentable subject-matter, Appellant submits that the claims recite a structure falling squarely within the subject matter acknowledged as patentable in *Lowry*. Appellant submits that the rejections of the pending claims based upon the Archibald reference do not present a *prima facie* case of obviousness and should be withdrawn.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Mark Joy', is written over a horizontal line.

Mark Joy, Reg. No. 35,562
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson
Chicago, Illinois 60601-6780
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: April 5, 2004



APPENDIX

The Pending Appealed Claims Consist of: 1-12, 14-44, 53-55 and 74-77.

1. A method for charging customers for use of software comprising the steps of:
establishing a use-based pricing scheme for a set of software modules;
distributing the set of software modules to a customer, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system;
monitoring customer use of the software modules; and
charging the customer according to use of the distributed software modules as determined during the monitoring step, wherein software usage is measured according to object instances created from the at least one object class.
2. The method of claim 1 wherein the customer creates a number of instances from a software module, and use of the software module is measured according to instances detected at a site of the customer during the monitoring step.
3. The method of claim 2 wherein instances created from a software module are periodically accessed to determine use during the monitoring step.
4. The method of claim 3 wherein the monitoring step comprises registering each day that an instance created from a software module is active; and wherein the charging step comprises charging the customer a daily rate for use of the software module.
5. The method of claim 1 further comprising the step of providing a demonstration mode for instances such that instances in the demonstration mode are executable at a customer site without charge.
6. The method of claim 1 further comprising maintaining a single agreement governing use of instances created from the set of software modules for an enterprise.
7. The method of claim 1 further comprising the step of monitoring a termination date for instances derived from a software module having a time-limited duration.

8. The method of claim 7 further comprising the step of issuing a warning in response to detecting an upcoming expiration date for an instance of a software module.

9. The method of claim 1 further comprising the step of maintaining an account for storing units of credit for a customer; and wherein said charging step comprises decrementing the customer's credit account by an appropriate number of units of credit based upon said monitoring step.

10. The method of claim 1 further comprising the step of generating a report summarizing use of software modules at the customer site.

11. The method of claim 1 wherein the charging step is based upon registered uses of a software module.

12. The method of claim 11 wherein the registered uses of a software module are measured according to execution of an instance created from the software module.

Claim 13 was previously deleted.

14. The method of claim 11 wherein the software module is an object class for creating an application engine object.

15. The method of claim 11 wherein the software module is an object class for creating a view engine object.

16. The method of claim 1 wherein the monitoring step comprises determining a time duration that an object instantiated from a software module is active.

17. The method of claim 1 wherein the monitoring step comprises registering execution of an instance that tracks throughput of a process.

18. The method of claim 1 wherein individual ones of the set of software modules are individually priced.

19. The method of claim 1 wherein the set of software modules includes at least a first software module supplied by a third party vendor and further comprising the step of:
compensating a third party vendor based upon a use by a customer of the first software module determined during the monitoring step.

20. The method of claim 1 wherein the distributing step comprises transmitting the set of software modules via a network connection.

21. The method of claim 20 wherein the network connection comprises an Internet connection.

22. The method of claim 1 comprising a step of reporting usage information to a software brokerage facility.

23. The method of claim 22 wherein the reporting step includes identifying the location of an instance created from a software module.

24. The method of claim 1 comprising the step of determining that a license manager has not reported to a software brokerage facility and in response registering a communication failure at a central licensing facility.

25. The method of claim 1 wherein the monitoring step includes storing use information in summary format in a database.

26. The method of claim 1 further comprising the step of issuing a re-ordering reminder to a customer.

27. The method of claim 1 wherein the software modules relate to industrial manufacturing automation software.

28. The method of claim 1 wherein the software modules relate to industrial manufacturing information software.

29. The method of claim 1 further comprising maintaining an agreement governing use of instances created from the set of software modules for an enterprise wherein the instances comprise both lifetime billed and use-based billed instances.

30. The method of claim 1 further comprising the step of providing configuration tools enabling a user to create customized instances from the software modules.

31. A method for vending software in the form of software modules via electronic commerce channels comprising the steps of:

maintaining an electronic commerce site including a software module selection interface, the software module selection interface enabling a customer to request a software module for use at a customer site, wherein the software module comprises at least one object class from which objects are instantiated on a customer system;

providing a software module management framework to the customer for installation at a customer site, wherein the management framework includes components for registering use of the software module at the customer site; and

charging the customer based upon registered use of the software module, wherein software module usage is measured according to object instances created from the at least one object class.

32. The method of claim 31 wherein the use of the software module comprises executing an instance created from the software module.

33. The method of claim 31 wherein the use of the software module comprises creating an instance from the software module.

34. The method of claim 31 wherein registering use of the software module provides a measure of throughput of an industrial process.

35. The method of claim 31 wherein the module management framework supports creation of instances from software modules at the customer cite having differing use modes including at least: a lifetime mode and a use-based mode, and wherein said method comprises the further step of registering execution of instances operating in the use-based mode.

36. The method of claim 35 wherein the use-based mode is measured in days and wherein an instance operating in use-based mode is registered each day in which the instance is executed.

37. A method for charging customers for use of software comprising the steps of:
providing a set of individually identifiable units of software comprising at least one object class from which objects are instantiated on a customer system;
individually pricing ones of the set of individually identifiable units of software;
authorizing use of the executable software; and
charging a customer based upon use of selected ones of the set of individually identifiable units of software, and wherein software usage is measured according to object instances created from the at least one object class.

38. The method of claim 37 wherein the authorizing step comprises transmitting a license file containing code enabling use by the customer of the executable software.

39. The method of claim 37 further comprising the step of:
integrating self-monitoring process software within the executable software; and
registering use of the executable software by the self-monitoring process.

40. The method of claim 37 wherein the executable software is industrial automation software.

41. The method of claim 37 wherein the self-monitoring process software comprises functions for informing the customer of a need to reorder credit to continue using the executable software.

42. A method for charging customers for use of software comprising the steps of:
first providing a set of software modules for software customers, wherein the set of software modules comprise at least one object class from which objects are instantiated on a customer system;

second providing a software licensing facility including a brokering facility through which software customers pay for software execution units, and wherein the brokering facility includes a set of software customer accounts; and

charging a software customer account a number of software execution value units based upon the value of software modules utilized by a customer, and wherein software module usage is measured according to object instances created from the at least one object class.

43. The method of claim 42 wherein the charging step is performed by an automated billing process.

44. The method of claim 42 comprising the further step of providing an on-line customer interface; and wherein the first providing step includes the step of providing a network interface enabling users to download software modules from a remote location.

Claims 45-52 were previously withdrawn from consideration in view of a restriction requirement.

53. A memory containing a software module structure facilitating automated distribution of software to customers, the software module structure comprising:

- a supplier identification;
- a product description;
- a billing definition; and
- an executable program module including at least one object class from which instantiated objects are rendered.

54. The memory containing a software module structure of claim 53 wherein the billing definition includes a usage rate.

55. The memory containing a software module structure of claim 54 wherein the billing definition includes a lifetime rate.

Claims 56-73 were previously withdrawn from consideration in view of a restriction requirement.

74. The method of claim 11 wherein the registered uses of a software module are measured according to creating an object instance from the software module.

75. The method of claim 42 wherein the use of the software module comprises executing an object instance created from the software module.

76. The method of claim 42 wherein the use of the software module comprises creating an object instance from the software module.

77. The method of claim 42 wherein registering use of the software module provides a measure of throughput of an industrial process.